Chapter 17
Secure Multi-Party Computation

Louis-Henri Merino and José Cabrero-Holgueras

17.1 Introduction

Secure Multi-Party Computation enables a group of parties to compute a function while jointly keeping their private inputs secret. The chapter discusses the definition of secure multi-party computation, its benefits and drawbacks, and its potential applications. It also discuss the trends in the field until 2025 and the challenges that need to be addressed for widespread adoption. Finally, the implementation possibilities for secure multi-party computation in Switzerland and the different deployment variations are discussed. The author provides recommendations for different markets and the need to consider deployment options.

17.2 Analysis

17.2.1 Definition

Secure Multi-Party Computation (MPC) enables a group of m mutually distrusting parties to jointly compute the outputs of a function $f(x_1, x_2, ..., x_m)$ where x_i is the ith party's private inputs without disclosing their private inputs [1]. The term "secure" indicates the latter property where the private inputs used for computation

L.-H. Merino (✉)
EPFL, Lausanne, Switzerland
e-mail: louis-henri.merino@epfl.ch

J. Cabrero-Holgueras
European Organization for Nuclear Research, Geneva, Switzerland
e-mail: jose@cabreroholgueras.com

V. Mulder et al. (eds.), *Trends in Data Protection and Encryption Technologies*,
https://doi.org/10.1007/978-3-031-33386-6_17

are kept secret from all other parties. Some MPC protocols allow for auditable computation allowing any party, including a party who did not participate in the computation, to verify the correctness of the result [2, 3].

A significant benefit of using MPC is that many of the constructed MPC protocols are information-theoretically secure, avoiding many of the problems involved with using cryptographic hardness assumptions. However, using MPC comes at the cost of performance (several orders of magnitudes slower), primarily due to MPC's high bandwidth requirements. Nonetheless, specialized MPC protocols can significantly enhance performance compared to generic MPC protocols; one prominent example is private set intersection [4]. A drawback of information-theoretic MPC protocols in comparison to MPC protocols that rely on hardness assumptions that their security guarantees are violated in the presence of a dishonest majority [5].

One particular case of multi-party computation is private set intersection (PSI). In this case, each party has a set of items, and the goal is to learn the intersection of those sets while revealing nothing else about those sets [6].

17.2.2 Trends

Virtually all organizations could see benefits from utilizing MPC as it enables mutually distrustful parties to cooperatively compute the output of a function that they all agree on without revealing their input. These parties may be distinct. (e.g., different healthcare providers aiming to collaborate to improve patient care but do not want to disclose patient data) or the same (e.g., an organization aiming to protect sensitive information by splitting this information across its multiple data centers, where each data center is a party to the MPC protocol).

Some notable MPC use cases are secure auctions [7], privacy-preserving network security monitoring [8], spam filtering on encrypted emails [9] and secure machine learning [10]. Another notable MPC application is distributed authentication where MPC can strengthen an organization's key server by splitting the critical server's functionalities across multiple servers; an adversary capable of compromising one or a threshold of critical servers will not be able to reconstruct the organizations' keys. Please refer to Chap. 13 for additional information on multi-party threshold systems. Unfortunately, factors such as a steep learning curve, unfamiliar mathematical notions, and a rapidly growing and evolving environment prevent easy exploitation of the technology by programmers and end users. To reach a widespread adoption of MPC, these issues must be addressed [11]. Application Programming interfaces (APIs) for secure multiparty computations are a promising technology to overcome these challenges. Another one are compilers.

17.3 Consequences for Switzerland

17.3.1 Implementation Possibilities: Make or Buy

An MPC solution consists of two major disciplines (distributed systems & cryptography), each with its challenges and it would thus require extensive efforts to design and implement a homemade MPC solution. The author then recommends purchasing an existing MPC solution for all markets (military, civil society, and economy) Nevertheless, he recommends different deployments as discussed in Sect. 17.3.2.

17.3.2 Variations and Recommendation

There are three MPC deployment variations: on-premise, hybrid, and cloud. For the military and maybe for civil society, the preferable setup is on-premise to prevent distributing private inputs to the software provider. To achieve the promises of MPC, an on-premise setup should require two or more independent data centers where each data center is considered a party to the MPC protocol. For civil society and the economy, the likely preferable option is a hybrid setup where the client's IT infrastructure and the software provider's IT infrastructure are each a party to the MPC protocol. The bandwidth between these two parties could be significant but may save the client from compartmentalizing their IT infrastructure. Cloud deployment allows for the complete outsourcing of the MPC solution where it is operated only on the software provider's IT infrastructure. This cloud deployment is likely the least expensive option.

17.4 Conclusion

In conclusion, MPC enables a group of mutually distrusting parties to compute an agreed-upon function using their own private inputs without revealing their private inputs to other parties. MPC can be used to secure and enable privacy-preserving applications from privacy-preserving network security to secure machine learning. Given the complexity of designing and implementing MPC protocols, enlisting an MPC provider is preferable, but clients should have flexibility over the type of MPC deployment: on-premise, hybrid, and cloud.

References

1. David Evans, Vladimir Kolesnikov, and Mike Rosulek. A Pragmatic Introduction to Secure Multi-Party Computation. April 2020.
2. Carsten Baum, Ivan Damgård, and Claudio Orlandi. Publicly Auditable Secure Multi-Party Computation. In Michel Abdalla and Roberto De Prisco, editors, *Security and Cryptography for Networks*, pages 175–196, Cham, 2014. Springer International Publishing.
3. Sanket Kanjalkar, Ye Zhang, Shreyas Gandlur, and Andrew Miller. Publicly Auditable MPC-as-a-Service with succinct verification and universal setup. *arXiv:2107.04248 [cs]*, July 2021.
4. Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu. Efficient Batched Oblivious PRF with Applications to Private Set Intersection. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 818–829, New York, NY, USA, October 2016. Association for Computing Machinery.
5. Shai Halevi, Yuval Ishai, Eyal Kushilevitz, and Tal Rabin. Best Possible Information-Theoretic MPC. In *Theory of Cryptography: 16th International Conference, TCC 2018, Panaji, India, November 11–14, 2018, Proceedings, Part II*, pages 255–281, Berlin, Heidelberg, November 2018. Springer-Verlag.
6. CSRC Presentation: A Brief Overview of Private Set Intersection | CSRC. https://csrc.nist.gov/presentations/2021/a-brief-overview-of-private-set-intersection, April 2021. CSRC | NIST.
7. Peter Bogetoft, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael Schwartzbach, and Tomas Toft. Secure Multiparty Computation Goes Live. In Roger Dingledine and Philippe Golle, editors, *Financial Cryptography and Data Security*, pages 325–343, Berlin, Heidelberg, 2009. Springer.
8. Martin Burkhart, Mario Strasser, Dilip Many, and Xenofontas Dimitropoulos. SEPIA: Privacy-Preserving aggregation of Multi-Domain network events and statistics. In *19th USENIX Security Symposium (USENIX Security 10)*, Washington, DC, August 2010. USENIX Association.
9. Trinabh Gupta, Henrique Fingler, Lorenzo Alvisi, and Michael Walfish. Pretzel: Email encryption and provider-supplied functions are compatible. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '17, page 169–182, New York, NY, USA, 2017. Association for Computing Machinery.
10. Jian Liu, Mika Juuti, Yao Lu, and N. Asokan. Oblivious neural network predictions via minionn transformations. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 619–631, New York, NY, USA, 2017. Association for Computing Machinery.
11. José Cabrero-Holgueras and Sergio Pastrana. Sok: Privacy-preserving computation techniques for deep learning. *Proceedings on Privacy Enhancing Technologies*, 2021(4):139–162, 2021.